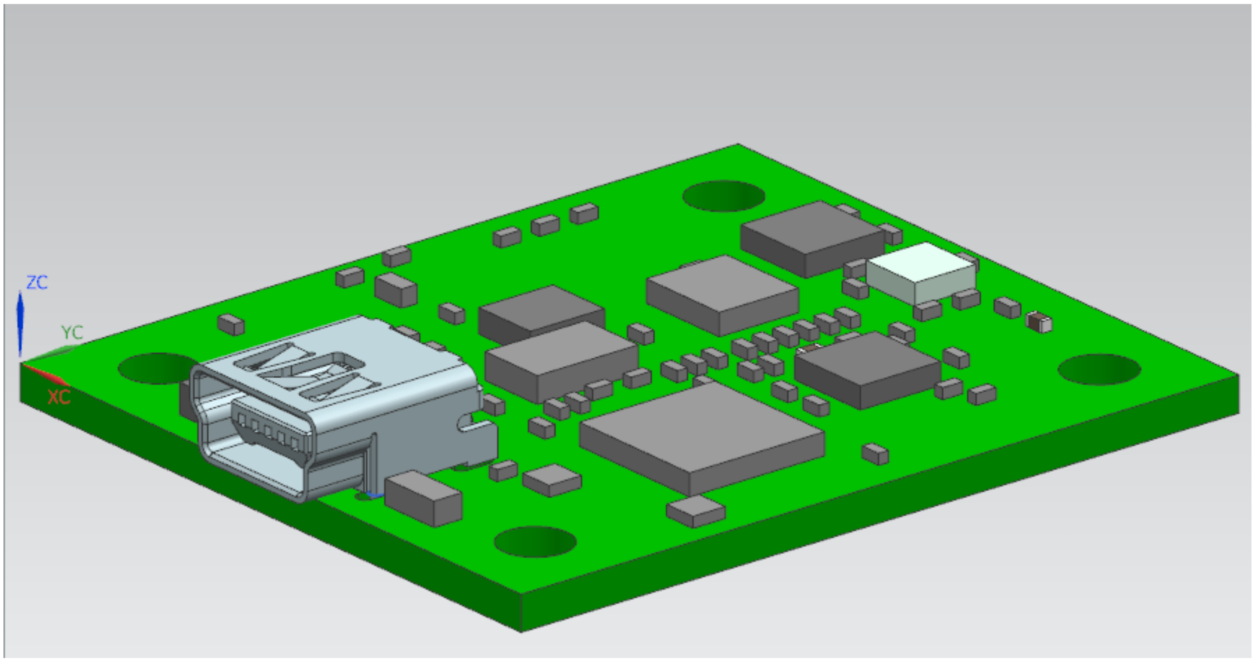
ENG 4350 6.0 FW16/17- Space Hardware

Phidget Spatial 1044 Digital Project



Keith Menezes

212839874

Prof. Hugh Chesser

Contents

Table of Figures

Summary

The purpose of this lab is to obtain angular measurements from the Phidget 1044 Spatial inertial measurement unit, to plot an antenna pattern.

Introduction

The Phidget 1044 contains a 3-axis accelerometer, a 3-axis gyroscope, and 3-axis magnetometer (compasses) to give the system 9 degrees of freedom. These microelectromechanical systems are combined to make an Inertial Measurement Units (IMUs). An Inertial Navigation System (INS) uses the output from an IMU, and fuses the information on acceleration and rotation with initial information about position, velocity, and attitude to then deliver a navigation solution with every new measurement. The first is the problem of sensor fusion, is how to combine the information obtained from the different sensors and obtain a good estimate of position and orientation. The second problem, is the of calibration of the sensors themselves. These are the principals I used for this project.

The purpose of this assignment is to demonstrate our ability to write a LabVIEW VI to measure the angular orientation of the Horn antenna under test. The intention is to use the orientation measurements from the IMU to produce an antenna pattern. Our LabVIEW program acquires angular measurements from the IMU and the user enters the antenna power at that elevation and/or azimuth change which is logged into the system.

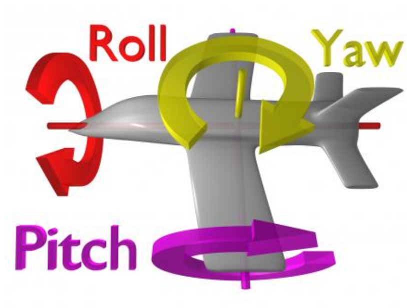


Figure 1: Roll, Pitch, and Yaw measurements will be used to determine orientation parameters

Sensors

1. Accelerometer sensors measure the difference between any linear acceleration in the accelerometer's reference frame and the earth's gravitational field vector.
2. In the absence of linear acceleration, the accelerometer output is a measurement of the rotated gravitational field vector and can be used to determine the accelerometer pitch and roll orientation angles.
    a. Realization – we can use accelerometer for the pitch angles.
    b. Simple vector algebra expressions are derived for computing the tilt of the accelerometer from vertical or the rotation angle between any two accelerometer readings.
3. Gyroscopes measure angular velocity, multiplied by time we can use that to find the angular direction.
    a. Obstacle – gyroscopic drift will cause errors in the measurements.

4. Magnetometers measure the magnetic field around the IMU.
    a. Realization – we can use the sensor integration of the 3-axis magnetometer to get a compass which can determine our heading with reference to magnetic North, this will be used for azimuthal measurements.
    b. Obstacle – electronics in the room will causes errors to the measurements, as well as being in the concrete building.

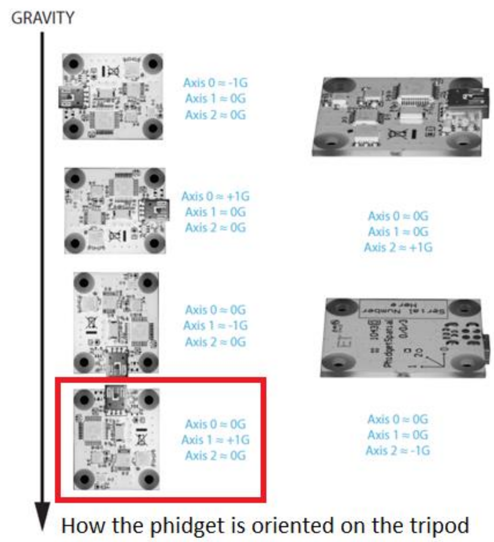


Figure 2: The orientation of the IMU on the Tripod is important to consider to determine the Home position

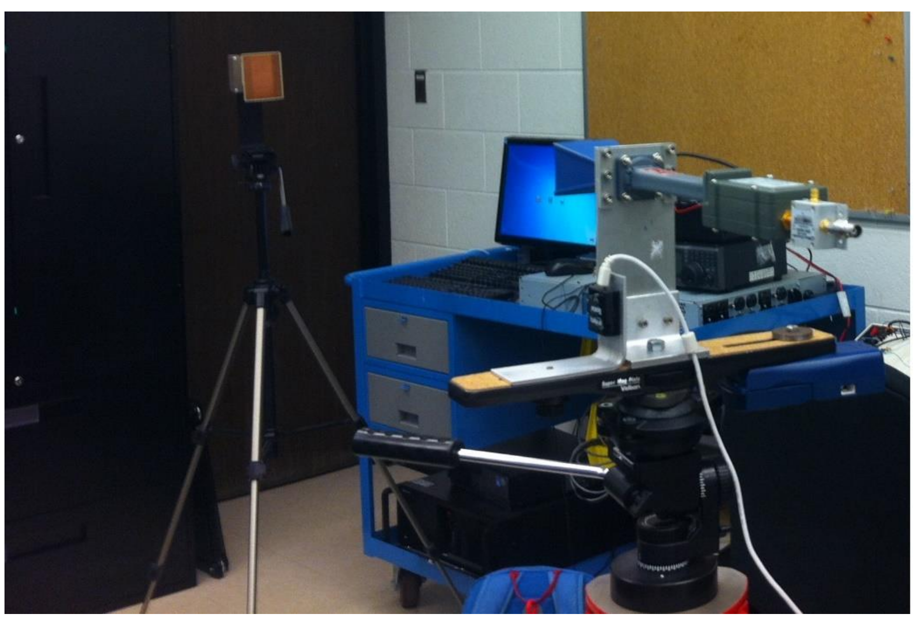


Figure 3: IMU on the Tripod

It can be seen in the figures above how the IMU is orientated with respect to the Horn antenna, which will be the basis of the static position.

Procedure

Calibration of the Compass

Compasses measure direction with respect to the 4 cardinal directions (North, East, South, West) with 0 degrees indicating straight North and 180 degrees indicating straight South.

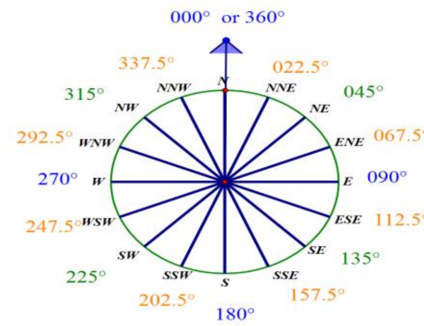
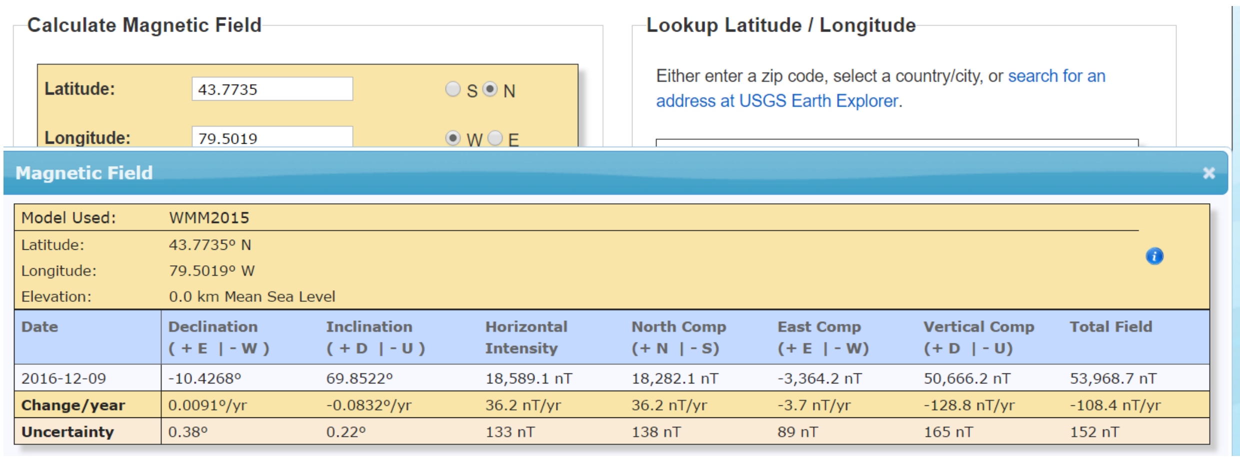


Figure 4: Relation to compass bearing and magnetic field measurements

First, to use the compass, we had to calibrate it and in order to run the calibration procedure, we used the C# code on the Phidget website. The C# code requires us to find out the magnetic field declination of our location in Petrie Science and Engineering building

Therefore we used the World Magnetic Model (WMM) or the International Geomagnetic Reference Field (IGRF) model to obtain the following parameters:



Date	Declination (+ E | - W)	Inclination (+ D | - U)	Horizontal Intensity	North Comp (+ N | - S)	East Comp (+ E | - W)	Vertical Comp (+ D | - U)	Total Field
2016-12-09	-10.4268°	69.8522°	18,589.1 nT	18,282.1 nT	-3,364.2 nT	50,666.2 nT	53,968.7 nT
Change/year	0.0091°/yr	-0.0832°/yr	36.2 nT/yr	36.2 nT/yr	-3.7 nT/yr	-128.8 nT/yr	-108.4 nT/yr
Uncertainty	0.38°	0.22°	133 nT	138 nT	89 nT	165 nT	152 nT

Figure 5: Magnetic Field Measurements from NGDC NOAA

Soruce: https://www.ngdc.noaa.gov/geomag-web/#igrfwmm

Next, we used the magnetic field value of 53.968.7nT (the total in Figure 5 above) and input it into the calibration program. Note that it wanted the field strength in Gauss, not nT like the website the conversion is: 1T = 10000 Gauss so you can divide by 1x10^5 to convert to Gauss.

Figure 6, to the right shows the completion of the calibration process where the values are also displayed to the screen just in-case they were reset by another group we could input those values again using the LabVIEW program or command-line interface.

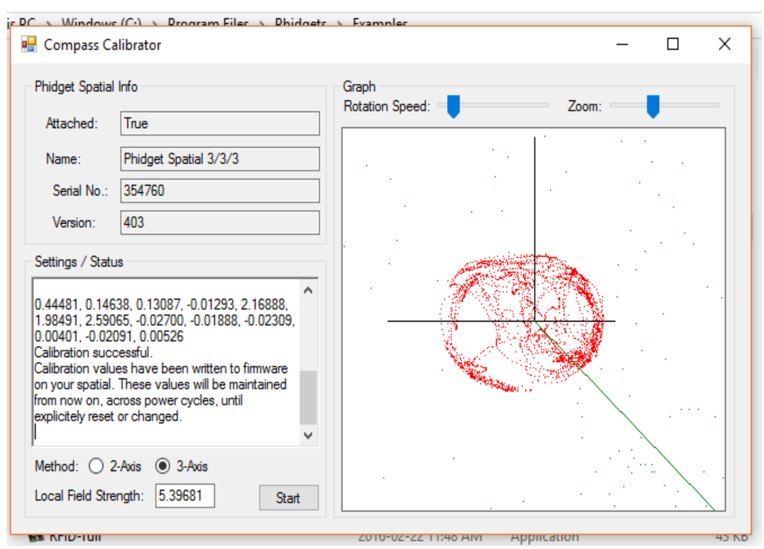


Figure 6: The calibration procedure involved rotating the IMU around in a circle about each axis while facing North, while it took measurements.

Methodology

Underlying Mathematics to obtain the Orientation Angles

Now we could start making measurements and computing the angles, the following equations were used:

Figure 7: Phidget orientation for reference

Sensor data:

$$acc0 = acclerometer\ measurement\ from\ the\ zero\ component$$

$$acc1 = acclerometer\ measurement\ from\ the\ one\ component$$

$$acc3 = acclerometer\ measurement\ from\ the\ two\ component$$

Note acc3 is actually for the two axis, this was already implemented in all the subVI's for Phidget's LabView code, so we left it like this.

$$mag0 = magnetometer\ measurement\ from\ the\ zero\ component$$

$$mag1 = magnetometer\ measurement\ from\ the\ one\ component$$

$$mag2 = magnetometer\ measurement\ from\ the\ two\ component$$

$$Roll = \operatorname{atan}2\left(\frac{acc1}{acc3}\right)$$

$$Pitch = \operatorname{atan}\left(\frac{acc0}{(\mathrm{acc1} * \sin(\mathrm{Roll}) + \mathrm{acc3} * \cos(\mathrm{Roll}))}\right)$$

To find azimuthal angle we checked the Phidget>Compass Primer where we found out how to calibrate the compass, then used the following source to get:

Gravitational, G and Magnetometer, M matrices where Z is the axis of the azimuthal calculation:

$$G = [acc0\ acc1\ acc3]$$

$$H = [mag0\ mag\ 1\ mag2]$$

$$G = \begin{bmatrix} Gx & Gy & Gz \end{bmatrix}$$
$$H = \begin{bmatrix} Hx & Hy & Hz \end{bmatrix}$$
$$Z = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}$$

$$B = G \times Z$$
$$A = G \times H$$

$$\cos\theta = \frac{(A \bullet B)}{|A| \bullet |B|}$$

Note these angles resulted in radians which were converted to degrees which can be seen below.

In Code form this is equivalent to:

```
float32 rollAngle;
float32 pitchAngle;
float32 Azimuth;
float32 Z1; float32 Z2; float32 Z3;
float32 Ax; float32 Ay; float32 Az;
float32 Bx; float32 By; float32 Bz;
float32 magA; float32 magB;
rollAngle = atan2(acc1,acc3);
pitchAngle=atan(acc0 / (acc1 *sin(rollAngle) + acc3 * cos(rollAngle)));
pitchAngle = pitchAngle * 180/pi;
Z1 = -1; Z2 = 0; Z3 = 0;
Bx = acc1*Z3 - acc3*Z2; By = acc3*Z1-acc0*Z3; Bz = acc0*Z2-acc1*Z1;
Ax = acc1*mag2-acc3*mag1; Ay = acc3*mag0-acc0*mag2; Az=acc0*mag1-acc1*mag0;
magA = sqrt(Ax**2+Ay**2+Az**2);
magB = sqrt(Bx**2+By**2+Bz**2);
Azimuth = acos((Ax*Bx+Ay*By+Az*Bz)/(magA*magB))*180/pi;
```

Once we had these reference values when the angle (either, elevation or azimuth) was changed we would have to do some subtracting to find our current position and the angle difference. The problem we ran into is that depending on the compass bearing angle, we may not be computing the angle change correctly if we simply subtracted the two values. Thus we tested for every case and solved the problem with some embedded if-statements for each case by comparing our 'initial compass bearing ICB' or 'reference azimuth' to the 'current compass bearing CCB'.

```
if(ICB>270  && CCB>0 && CCB<(90+(ICB-270)))
CCB = CCB-ICB+360;
else if (ICB>180 && ICB<270 && CCB>0 && CCB<(ICB-180))
CCB = CCB-ICB+360;
else if (ICB>90 && ICB<180 && CCB<360 && CCB>(360-(180-ICB)))
CCB = CCB-ICB-360;
else if (ICB>0 && ICB<90 && CCB>(180+ICB) && CCB<360)
CCB = CCB-ICB-360;
else
CCB = CCB - ICB;
```

Figure 8: Problem solving azimuthal/compass bearing issues

Now that we have our angular orientation parameters being computed and we can compare the two values to give us reasonable results we implemented the user-interface and functionality necessary to obtain the antenna pattern plot.

User Interface with LabVIEW

Essentially, we wanted to make a clean and user friendly interface for the client. We broke it down into the easiest control we could think of:

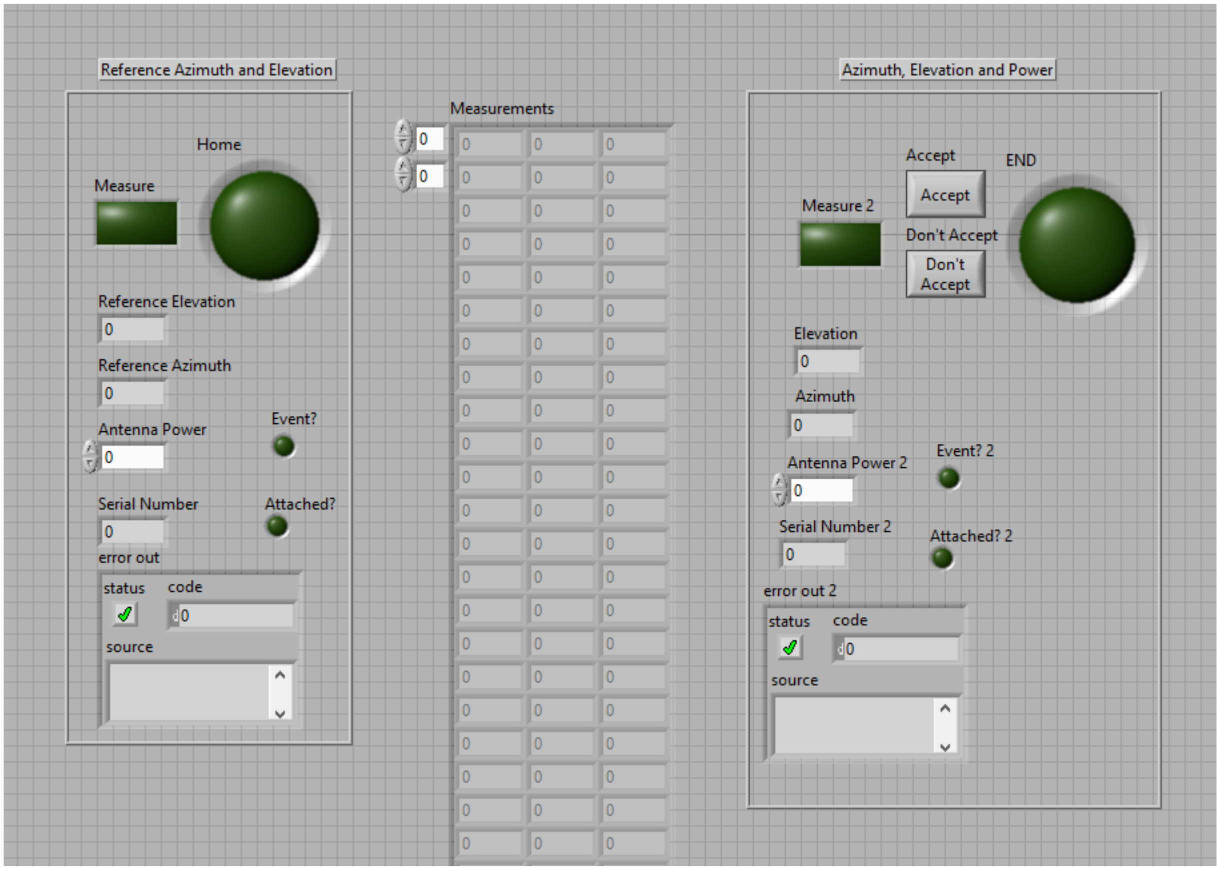


Figure 9: User Interface for our VI

The VI presents buttons to the user with buttons to measure orientation, set that position as home, accept/reject the measurement, and end the program. The flow of the program is very simple, the client can measure orientation angles with respect to the current orientation, accept or reject the measurements (just in case of error, or the antenna wasn't moved in the right direction) where then it is written to the table. This can be repeated for 50 values (or more if we extend the table/memory limit) then the user presses end, to close and terminate the program. And all the measurements are saved to a file indicated by the client.

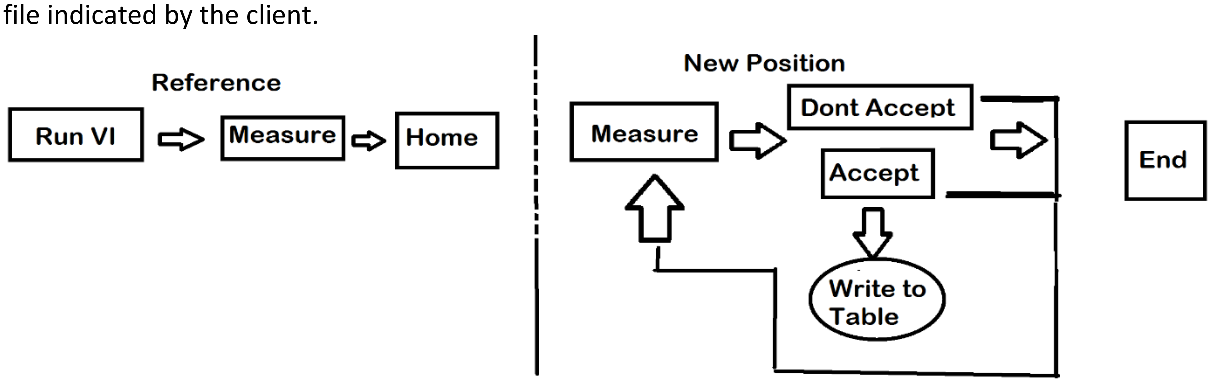


*Figure 10: User interface methodology, *note* end can be pressed at any time once Home is clicked*

Step 1: Execute the Code

Step 2: Orientate the instrument.

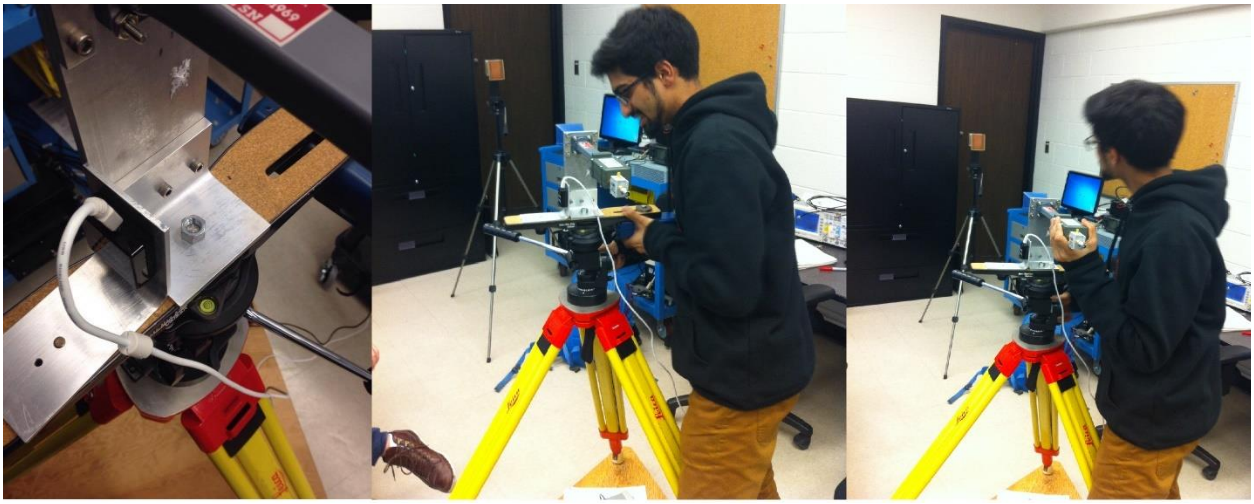


Figure 11: Orientating the Antenna and leveling it for the Home position

Step 2: Click Measure until at desired Home location. Then Click Home.

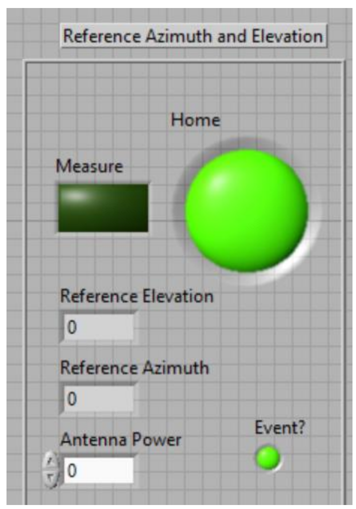


Figure 12: Click Home once at desired initial position

Step 3: Move the Antenna and input the antenna power, then Click Measure to see the change.

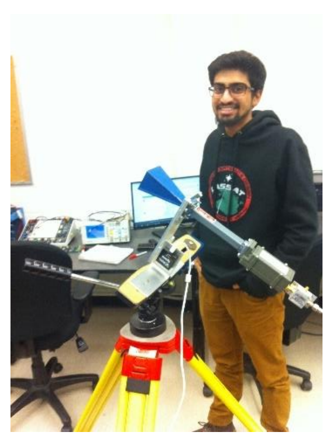


Figure 13: Changing the orientation of the Horn Antenna

Step: 4: Accept or Deny the Measurement.

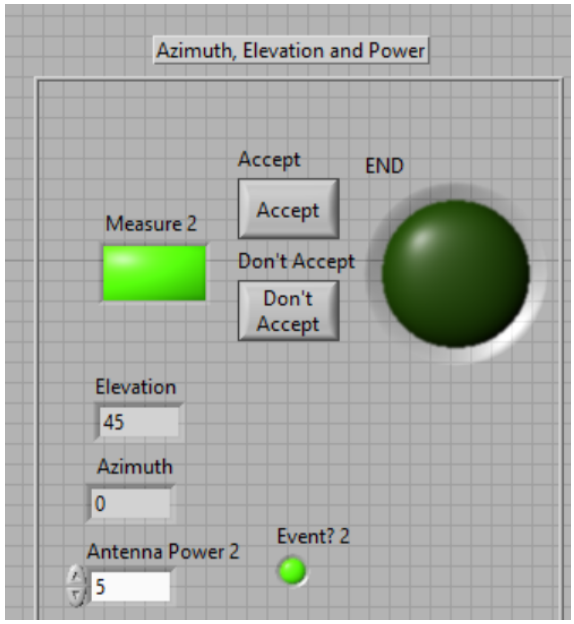


Figure 14: Accept or Deny the Measurement

Step 5: Step 3 and 4 are repeated until the user is satisfied

Step 6: Click END.

Code Explained

Once the code is executed there is a check to see if the Phidget is connected to the computer. If satisfied it enters the while loop where the Phidget Spatial Even is used to get sensor data upon clicking 'Measure'. The while loop terminates once 'Home' is pressed. The formula node is what computes the angles mentioned in the Methodology section.

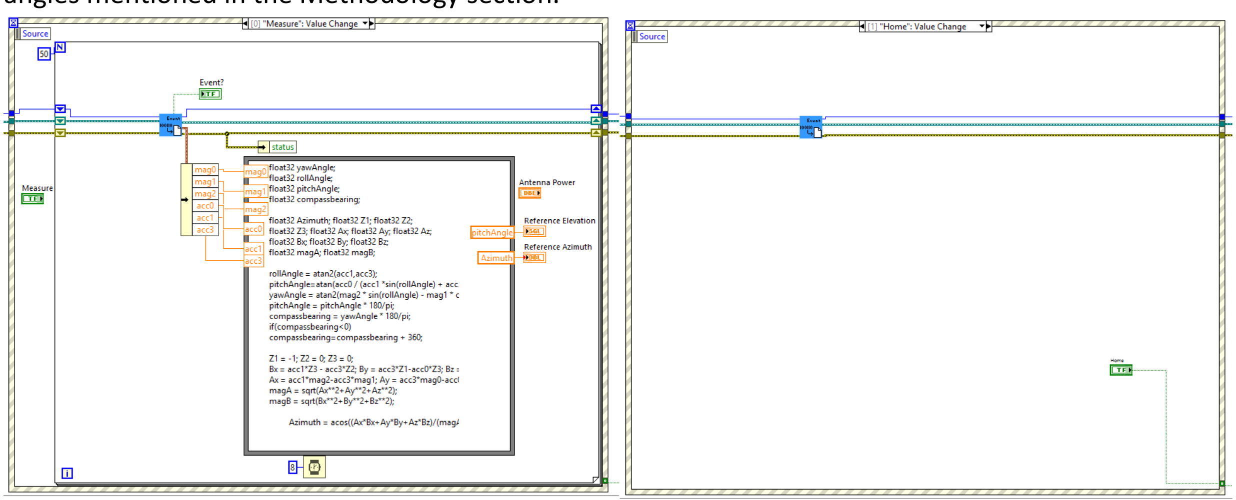


Figure 15: Measure/Home Event Structures

The reference azimuth and reference elevation are stored into local variables, to be used in the next part of the code. Note that the azimuth angle is between 0 and 180 for both clockwise and counterclockwise measurements while the elevation angle goes from -180 to 180, where down is negative and up is positive.

Now that we have the 'Home' position, we move to the second part of the code (right side of the user interface) where once the user presses 'Measure' we take the 'initial compass bearing' ICB and compare it to our 'current compass bearing' CCB. We also do the same for the reference elevation 'pitch angle' and new elevation angle or pitch angle.

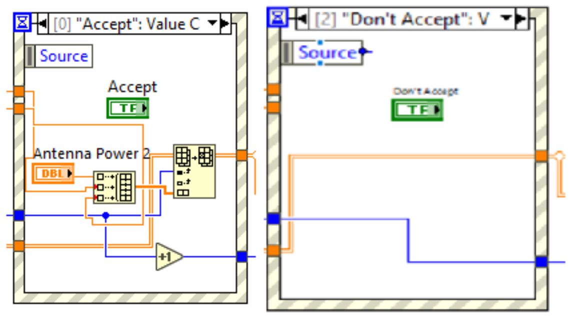


Figure 16: Accept or Reject the Measurement

The user is then prompted to accept or reject the measurement where if accepted it will be written to the table.

The measurement, accept/reject, process is repeated for 50 more times or until the user presses stop, where they are prompted to write the tabled data of accepted values to a file (.csv) to the location of their choice on their computer.

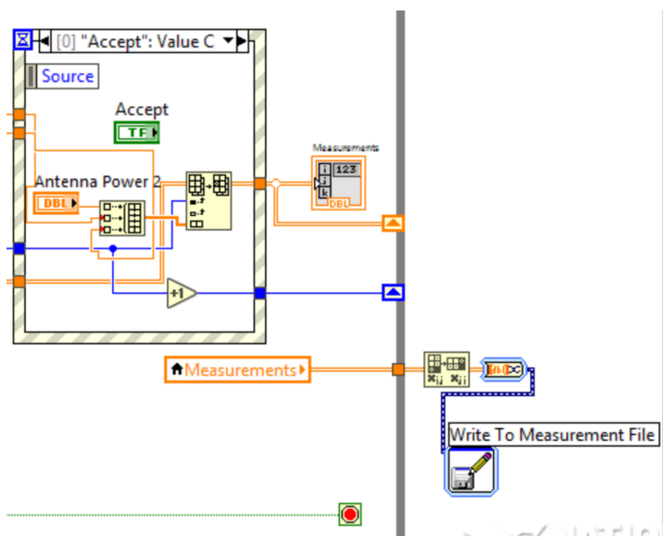


Figure 17: Writing the accepted measurements to a file, upon hitting END

Testing

To the right is a table of test data that we got from our LabVIEW program. For the 0-7 measurements, we were testing the azimuthal angle and for the 8-14 tests we were testing the pitch angle. When the Horn antenna is pointed at the floor (-82°) there is a change in the azimuthal angle and the same occurs for the Horn antenna being pointed at the ceiling (88°). This also has to do with how the compass works but is satisfactory for this project because we are only measuring one axis at a time.

Power	Elevation	Azimuth
0	0	0
1	-0.82551	157.6773
2	-1.62443	124.7275
3	-1.24497	97.787
4	-1.5453	59.56061
5	-1.75326	21.96831
6	-1.54465	58.35671
7	-0.99986	126.8147
8	-1.00047	165.9088
9	-27.6349	165.4516
10	-53.2804	165.1639
11	-82.986	19.25591
12	16.45825	164.6932
13	41.42579	164.202
14	88.54105	172.8494

Figure 18: Output data from our VI, viewed in Excel

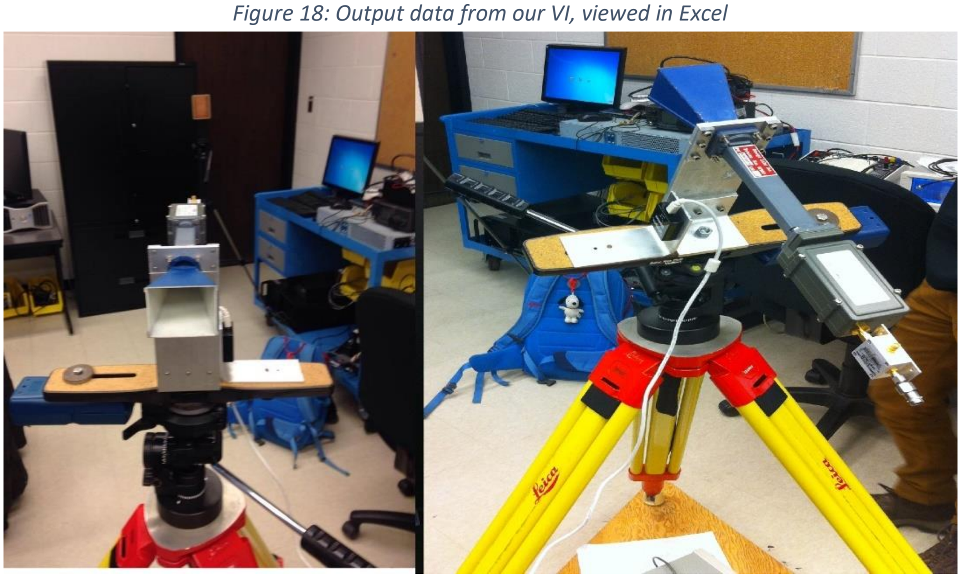


Figure 19: Testing our Elevation and Azimuth angles

Bonus

I took these values and then input them into my polar plot VI. My polar plot VI (Appendix), when executed asks for a file, with the antenna measurements just as I have them above. The problem, was that I was unable to resolve any information of what the polar plot means, this is probably because there are so few data points, and we have not yet learned in class the meaning of the antenna pattern plot. In the picture below I wanted to have it plot in 2D (for each axis, elevation, and azimuth) and in 3D as well to show the relative magnitude of power. The red line in the image to the right under 'new picture' is the data points from above.

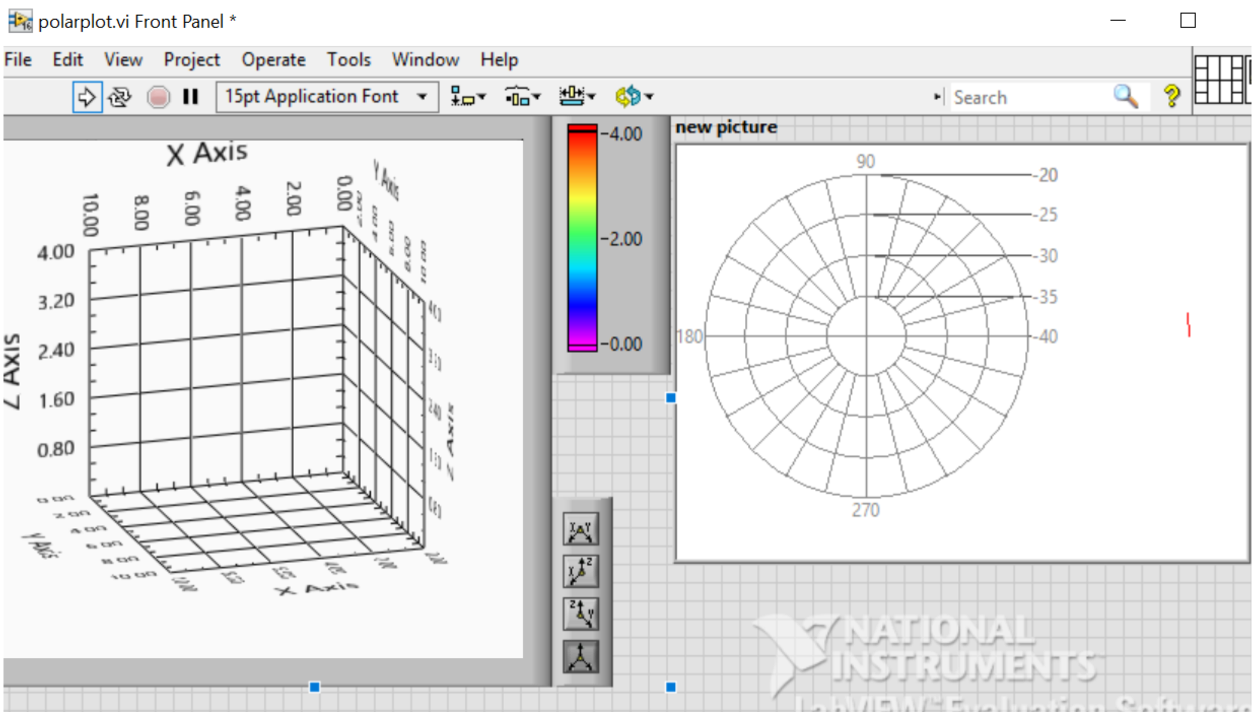


Figure 20: The polar plot has some bugs, the red line which is plotted on the right (2D) is not at the origin

Discussion

Elevation

To obtain the elevation measurements we used the recommendation from the Phidget documentation so we get an accurate solution of the elevation. As discussed, it's the best to use when measuring tilt where the system is static, and we require high resolution and low noise. From the data sheet we could find the accelerometer to have a resolution of +/- 76.3 µg, white noise of 280 µg, and minimum drift of 40.6 µg.

Precision Accelerometer

Acceleration Measurement Max	± 2 g
Acceleration Measurement Resolution	76.3 µg
Acceleration Bandwidth	497 Hz
Accelerometer White Noise σ	280 µg
Accelerometer Minimum Drift σ	40.6 µg
Accelerometer Optimal Averaging Period	398 s

Figure 21: Precision of the Accelerometer

Accelerometer Noise

Accelerometer chips will always be effected by some amount of noise, usually caused by thermal and mechanical fluctuations inside the sensor. Noise levels will be different for each axis, because each orientation of sensor is built slightly differently (for example, it is much more difficult to get low noise from the vertical axis because the sensor is a flat piece of silicon. Two of the major types of noise are white noise and random walk.

Accelerometer White Noise and Random Walk (Drift)

White noise is the short-term noise that is contributed to from a number of internal and external factors. For example, when an accelerometer is stationary on our desk, it might read 1.00g one sample, 0.99g another sample, and 1.02g in yet another sample. Luckily, white noise is usually fairly consistent which means it can be mitigated quite effectively if you average multiple samples together. With the Phidgets Inc. accelerometer, we set data rate to much slower than its maximum so it automatically averages as many samples as it can within that time frame for each value.

Also, there could be noisiness in the lab environment- For example, students jumping or moving around near the antenna, will easily be noisier than the accelerometer.

Often called drift, random walk is the long-term noise that causes samples to gradually become further and further away from their true values.

These sources of error should not affect our calculations because that is very precise, considering we are only measuring fractions of g's. Taking this into account we don't expect much error in the antenna pattern for the elevation plot.

Azimuth

To obtain the azimuthal angle the accelerometer and magnetometer were used. The magnetic field is represented as a vector. If the spatial is completely level with respect to gravity, then we can calculate magnetic bearing easily by looking at the x and y components of the vector, which simply represents an angle between 0 and 360 degrees on the unit circle.

However, when the board is not level, the magnetic field vector components in each axis will be skewed, or is other words, what we need to know is the magnetic field with respect to the earth frame, and what we have is the magnetic field with respect to the Phidget, which is rotated with respect to the earth. What needs to be done is to rotate the magnetic field vector back into the earth frame, we can then use the x-y components to again determine bearing. The easiest way to do this is to find the board pitch and roll angles using the accelerometer, and then build a rotation matrix using these angles to rotate the magnetic field vector into the earth frame (ie aligned with gravity).

A compass is a navigational instrument that is used to determine direction relative to the surface of the earth. Compasses measure direction with respect to the 4 cardinal directions (North, East, South, West) with 0 degrees indicating straight North and 180 degrees indicating straight South, as mentioned before.

Electronic compasses work in exactly the same way as any other compass: They detect the Earth's magnetic field and respond to it. The difference is that an electronic compass uses a magnetometer to detect the field as opposed to a small magnet. This allows them to be much more accurate and allows them to respond more quickly to changes in direction than a traditional compass ever could.

Small compasses found in clocks, mobile phones, and other electronic devices are solid-state compasses, usually built out of two or three magnetic field sensors that provide data for a microprocessor. The correct heading relative to the compass is calculated using trigonometry. Often, the device is a discrete component which outputs either a digital or analog signal proportional to its orientation. This signal is interpreted by a controller or microprocessor and used either internally, or sent to a display unit. The sensor uses highly calibrated internal electronics to measure the response of the device to the Earth's magnetic field.

We calibrated the compass to handle the magnetic field characteristics in our particular laboratory (latitude/longitude of the Petrie Science and Engineering Building, and estimated the height of the lab by estimating the height of the 3rd floor above ground level) as well as disturbances caused by a more localized magnetic field such as speaker magnets. To calibrate the compass effectively we determined the Earth's magnetic field strength at the lab, as seen in the section above.

The most common problem compass users can expect to face is interference. Interference will generally cause the compass to behave poorly. Usually this means inconsistent, fluctuating, or inaccurate readings. Luckily most forms of interference are fairly easy to deal with and are resolved with the calibration procedure.

Hard Iron Interference

Hard Iron interference is caused by static magnetic fields associated with the metal near the IMU. This can include any minor (or major) magnetism in the metal chassis or frame of the antenna stand. This interference pattern is unique to the vehicle but is constant. Since we have our compass in an enclosure that is held together with metal screws even these relatively small amounts of ferromagnetic material can cause issues. If we consider the magnetic data circle, hard iron interference has the effect of x/y/(z) in the case of 3 axes) shifting the entire circle away from the origin by some amount. The amount is dependent on any number of different factors and can be very large. The important part is that this shift is the same for all points in time so it can be calibrated out very easily with a numeric offset which is taken care of by the calibration process, thus won't affect the polar plot.

Soft Iron Interference

Distortion of the Earth's magnetic field causes soft iron interference due to materials in the vehicle's construction. It's like electricity, the magnetic field is looking for the easiest path to get to where it is going. Since magnetic fields can flow more easily through ferromagnetic materials than air, more of the field will flow through the ferromagnetic material than you would expect if it were just air. This distortion effect causes the magnetic field lines to be bent sometimes quite a bit. Note that unlike hard iron interference which is the result of materials which actually have a magnetic field of their own, soft iron interference is caused by non-magnetic materials distorting the Earth's magnetic field. This type of interference has a squishing effect on the magnetic data circle turning it into more of an ellipsoid shape. The distortion in this case depends on the direction that the compass is facing. Because of this the distortion cannot be calibrated out with a simple offset, more complicated math will still let the compass account for this type of interference though.

Declination

The magnetic declination calculator accounted for this. The North magnetic pole shifts and is not the same as true north over time. Since we are only using over such a finite period it won't affect the polar plot but, has been solved for with the calibration process.

Compass

Magnetic Field Max	5.5 G
Compass Resolution	3 mG
Compass White Noise σ	1.1 mG
Compass Minimum Drift σ	78 μG
Compass Optimal Averaging Period	1443 s

Figure 22: Precision of the Compass

Also, the random walk and white noise also occur for the compass. It has an resolution of 3mG, white noise of 1.1mG, and minimum drift of 78μG. The magnetic field of Earth is around 0.5G which means the error is more than the accelerometer but still within a reasonable amount to give accurate orientation for our purpose.

Despite the accumulation of all these errors in the IMU, the experimental results were very close to what we would expect (i.e. if we rotated 180° we would get around +/- 5° or 10°), plus we have the measure again option, which enabled us to get the most precise value.

There are many other ways to get the antenna orientation. The easiest is in the Phidget documentation which states Pitch = arctan(acc1), acc1 or relevant axis. Then for Azimuth, as mentioned in the lab the accelerometer can be used by offsetting the antenna on the tripod (lifting one of the legs) to get acceleration changes due to gravity while measuring the yaw angle or azimuth. This gives a component of gravity g*sin(θ), where θ is the degree of offset in the tripod legs from level. We found this to be less practical and a lot more effort for the user to make measurements. Also, the measurements were not as precise.

Another method would be to use the sensor measurements from the gyroscopes but it is also difficult to get an accurate measurement from the gyros as well as integrate over the time duration of the measurement.

$$Pitch = Pitch + \Delta Pitch * \Delta t \quad (1)$$
$$Roll = Roll + \Delta Roll * \Delta t \quad (2)$$
$$Yaw = Yaw + \Delta Yaw * \Delta t \quad (3)$$

Figure 23: Equations to measure Pitch, Roll, Yaw from Gyros

Another method of obtaining the orientation would be to use the markings on the tripod-mount for the antenna. These markings go in a full circle around where the antenna rotates both vertically and horizontally. These markings can indicate what your current orientation is if, you know where the markings were when you started.

Aligning the Antenna

We made it so the alignment is very simple. Figure 11 shows the bulls-eye level, which is used for many applications, surveying and carpentry. When the bubble is inside the circle it is level. The only inaccuracies would be due to temperature, but the room of the lab stays fairly consistent. The antenna should be as close to level as possible to begin the elevation adjustments. Then the user can use the spin adjustment of the tripod to precisely align the antenna to its mount axis. There is a screw on each of the elevation and azimuthal axes which can be loosened to make the adjustments. The antenna is aligned with the mount axis when the bubble is centred within the circle of the bulls-eye level. When the bubble is centred, the user should tighten the screw and make adjustments in place with the antenna aligned to the mount axis.

Filtering

This is when noise filtering and prediction/forecasting techniques are used to determine calibration parameters so that the absolute accuracy of the system does not drift. Techniques like smoothing and filtering are used to get more precise values from the sensors in the IMU. The most common filtering technique in INS is the Kalman filter which is a linear quadratic estimation algorithm. The Kalman filter keeps track of the estimated state of the system and the variance of the estimate, to obtain precise control and trajectory optimization. The Kalman filter could've been added to each sensor measurement to increase the accuracy if determined to not be precise enough.

Conclusion

In conclusion, this project was very educational and practical. I was able to use my knowledge of Unmanned Aerial Systems to determine what sensors in the Phidget would help get the orientation parameters necessary for completion of the lab. I was also able to develop my proficiency with the LabVIEW program as well as work with others where I was able to educate my team about what I've learned, what progress I've made, as well as also learn from them. I learned how to create a user interface in LabVIEW as well as take into consideration user experience while learning more about accelerometers, gyroscopes, and magnetometers (i.e. sensor measurements).

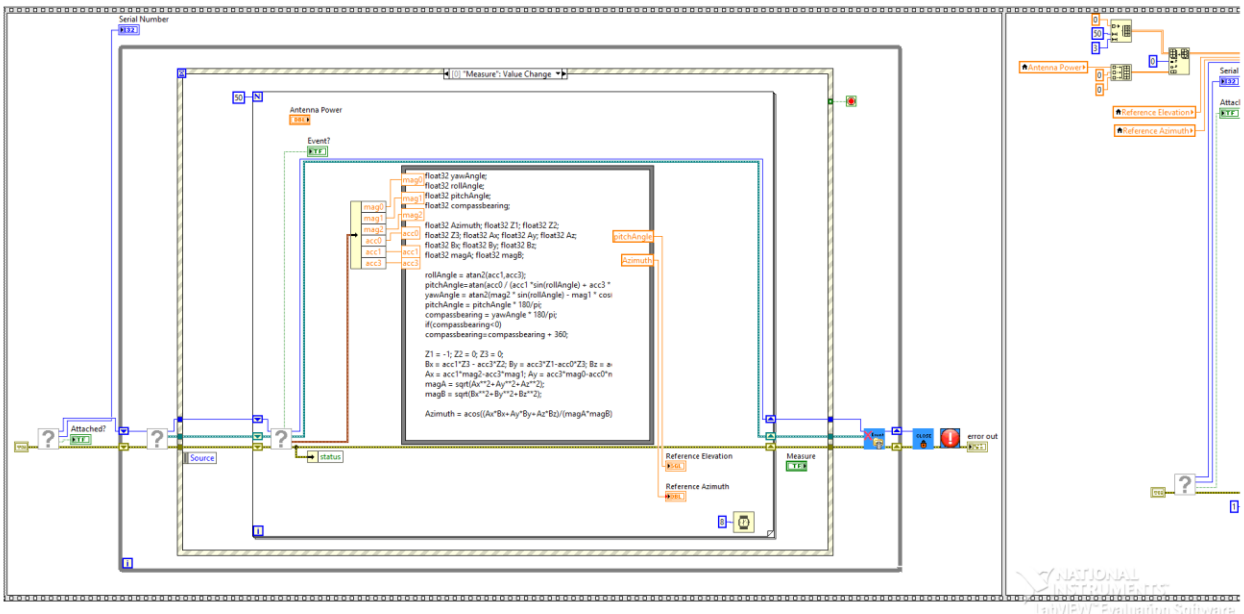
To improve our LabVIEW code I would implement the polar plot code into the front panel of the main code. Also, learn what the antenna pattern means, and implement it such that a realization of the antenna pattern can be made. Also, to improve the code we could have had just 4 buttons: 'Measure', 'Accept', 'Home', and 'End'. This would make it even easier for the user such that they could measure and change the home as well as keep writing that to the same file. The last improvement that could've been made to the code would be that it was still a bit buggy by the end, sometimes it would lag or freeze if the wrong button was pressed.

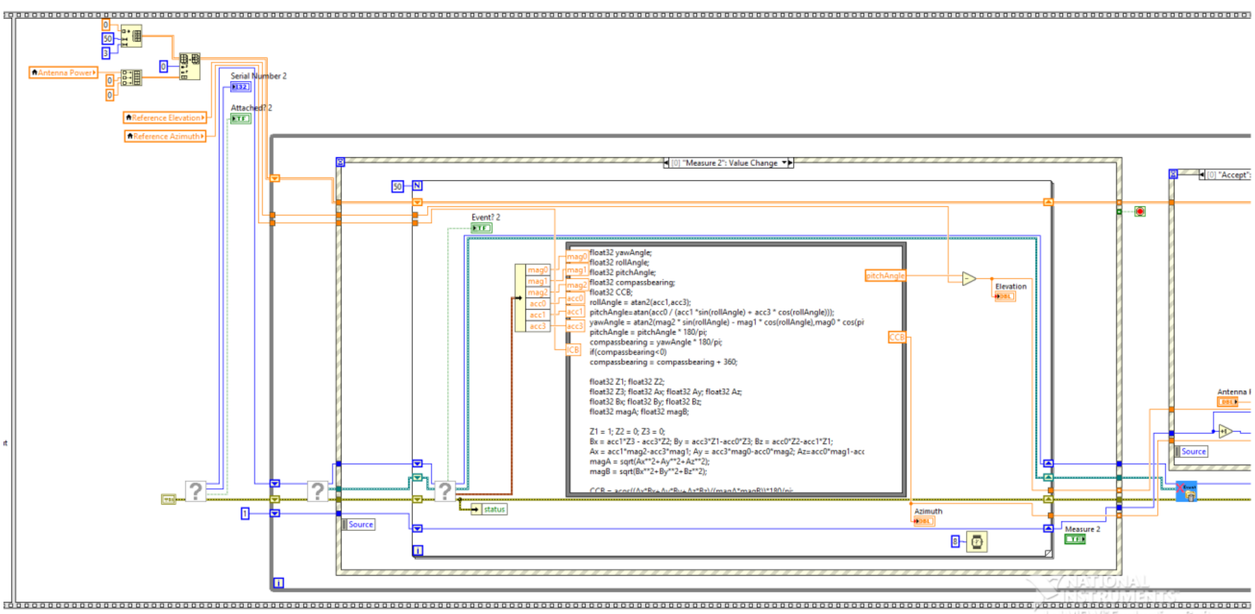
An improvement to our implementation would be to have the azimuth go from 0-360° rather than 0-180° as well as be able to tell counterclockwise from a clockwise rotation. The same goes for the elevation angles, if the antenna mount wasn't restricted to 90° movements up and down (i.e. it couldn't get elevations from 0-180° due to the mount in the way) we could obtain similar antenna plot to the ones on the lab manual.

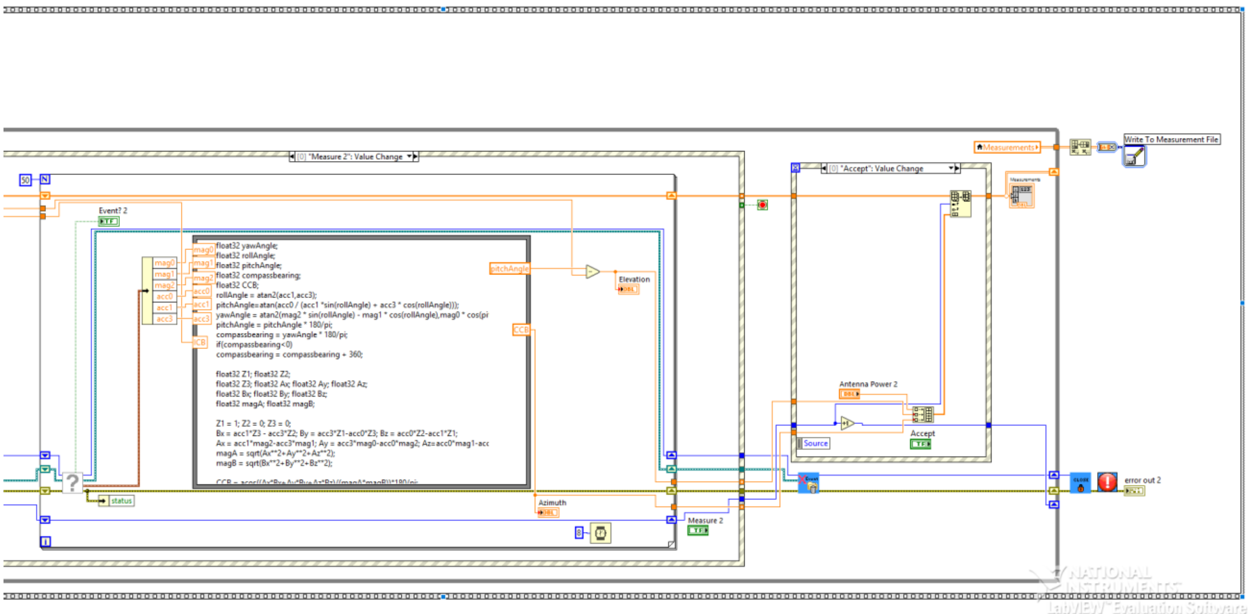
I really enjoyed this project, and very much preferred it instead of the test because I was able to apply my hands-on skills to produce a successful product.

Appendix

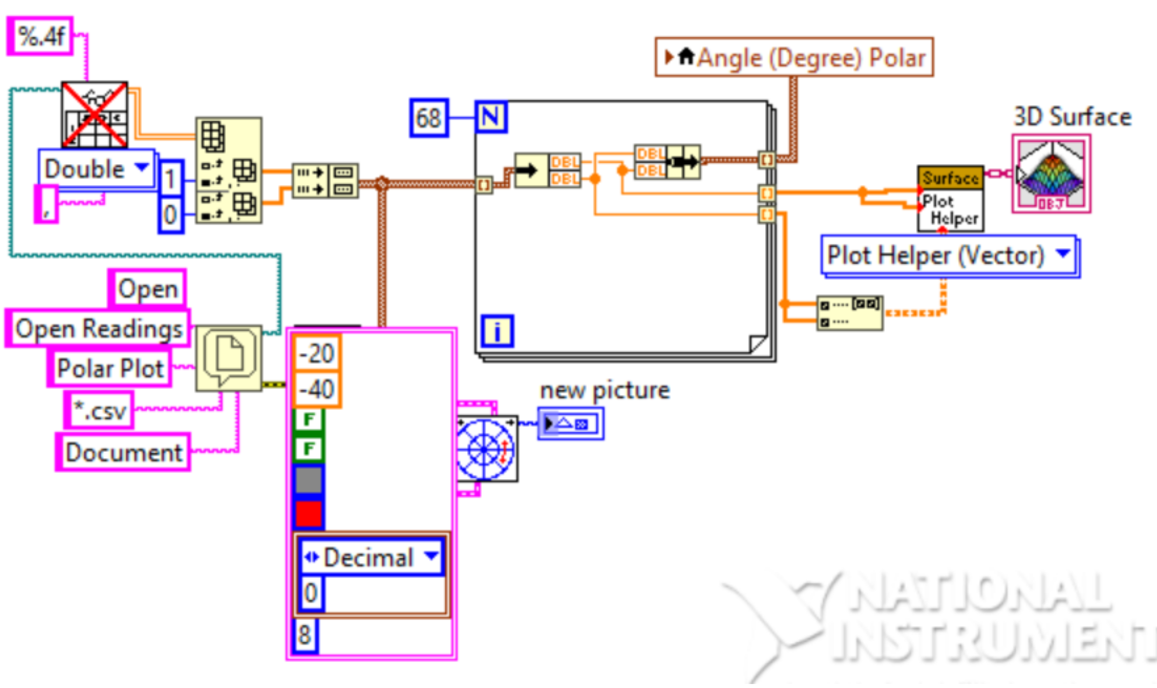
Code – Phidget

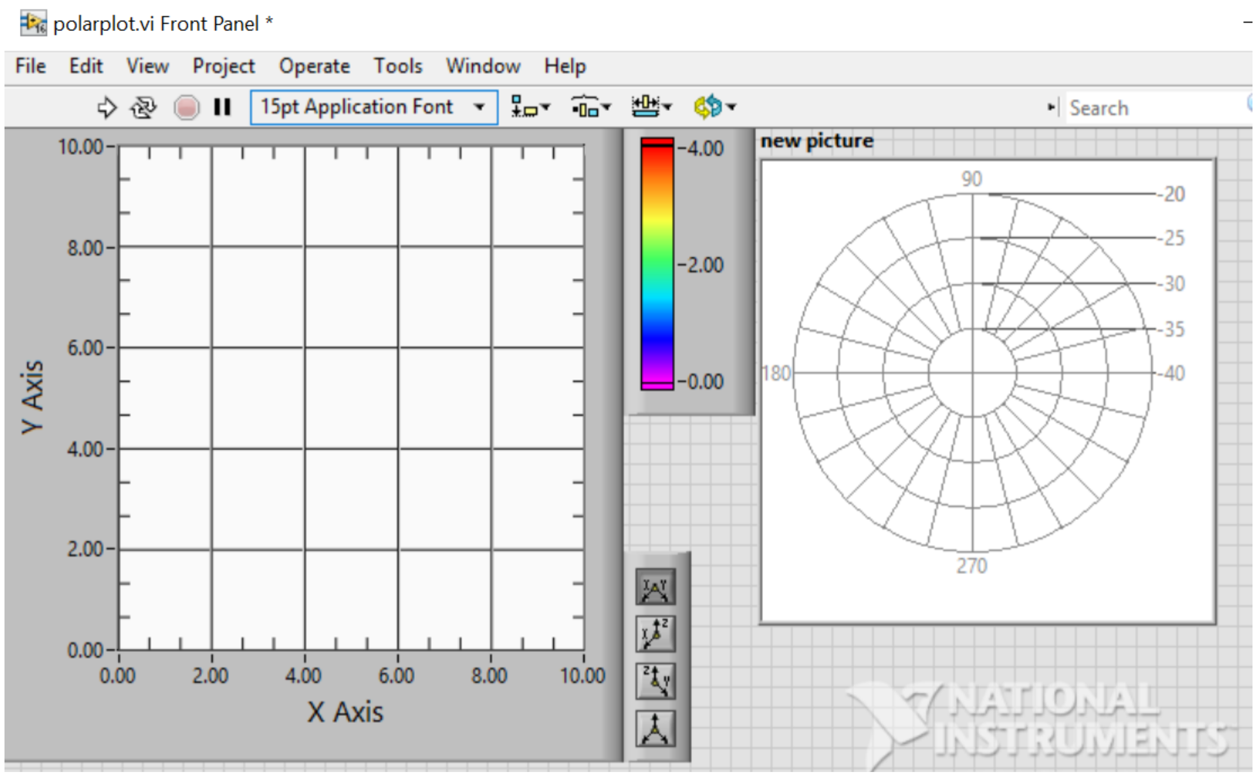






Code – Polar Plot




polarplot.vi Front Panel *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
10.00
8.00
6.00
4.00
2.00
0.00
Y Axis
0.00 2.00 4.00 6.00 8.00 10.00
X Axis
-4.00
-2.00
-0.00
new picture
90
180
270
-20
-25
-30
-35
-40
NATIONAL INSTRUMENTS